

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of claims:

1. (original) A method for cache management comprising:

requesting database lock of a named cache;

locking the named cache and providing an indication that the named cache is

locked;

acquiring a local lock of the named cache at a local node;

generating a timestamp corresponding to the local lock;

invalidating a cache item of the named cache in the local node;

releasing the local lock of the named cache;

sending a message to a remote node identifying the cache item of the named

cache;

receiving acknowledgment from the remote node;

sending an update of the cache item of the named cache;

updating the named cache; and

releasing the database lock of the named cache.
2. (original) A method according to claim 1, further comprising:

acquiring a local lock of the named cache at the remote node.
3. (original) A method according to claim 1, further comprising:

invalidating the cache item of the named cache at the remote node.

4. (original) A method according to claim 1, further comprising:

acknowledging the message identifying the cache item of the named cache.

5. (original) A method according to claim 1, further comprising:

releasing the local lock of the named cache at the remote node.

6. (original) A method according to claim 1, wherein locking the named cache

locks the named cache for all nodes.

7. (presently amended) A method for cache management in a distributed computer system comprising:

identifying a cache miss of a cache item;

requesting a read lock of a named cache, the named cache including the cache item;

read locking the named cache;

requesting the cache item from a master locking database;

receiving the cache item; and

releasing the read lock of the named cache.

8. (presently amended) A method according to claim 7, further comprising:

sending an indication that the named cache is read locked.

9. (presently amended) A method according to claim 8, further comprising:
sending the cache item from the master locking database.

10. (presently amended)A method for cache management in a distributed computer system comprising:
identifying a cache miss of a cache item;
requesting a read lock of a global database, the global database including the
cache item;
read locking the global database;
requesting the cache item from a master locking database;
receiving the cache item; and
releasing the read lock of the global database.

11. (presently amended) A method for cache management in a distributed computer system comprising:
determining that a predetermined event has occurred;
requesting a read lock of a named cache;
requesting a timestamp;
receiving an indication of a read lock of the named cache;
receiving a timestamp;
comparing the received timestamp with a previous timestamp;
responsive to the comparison, performing a predetermined action; and

releasing the read lock of the named cache.

12. (original) A method according to claim 11, wherein the predetermined action comprises storing the received timestamp.

13. (original) A method according to claim 11, wherein the predetermined action comprises requesting an update of the named cache.

14. (original) A method according to claim 11, wherein the predetermined action comprises receiving an update of the named cache.

15. (original) A method for cache management comprising:

- determining at a local node that an update is required for a cache item in a named cache;
- sending a request for a database lock of the named cache from a local node to a cache manager;
- receiving the request for a database lock at the cache manager;
- locking the database of the named cache;
- sending an indication that the database of the named cache is locked from the cache manager to the local node;
- receiving the indication that the database of the named cache is locked at the local node;
- acquiring at the local node a local write lock of the named cache;

sending a timestamp from the local node to the cache manager;
receiving the timestamp at the cache manager; updating a lock table with the
timestamp;
invalidating at the local node the cache item of the named cache;
releasing the local write lock of the named cache;
sending a message to invalidate the cache item of the named cache from the local
node to a second node;
receiving the message to invalidate at the second node acquiring at the second
node a remote write lock of the named cache;
invalidating at the second node the cache item of the named cache;
sending an acknowledgment of the message to invalidate from the second node to
the local node;
releasing the remote write lock of the named cache at the second node;
receiving the acknowledgment of the message to invalidate at the local node;
determining that no further acknowledgments of the message to invalidate are
expected at the local node;
sending an update of the cache item of the named cache from the local node;
receiving the update of the cache item and updating the cache item; and
releasing the database lock of the named cache.

16. (presently amended) A method for cache management in a distributed
computer system comprising:

identifying a cache miss of a cache item at a local node;

sending a request for a read lock of a named cache from the local node to a cache manager;

receiving the request for a read lock of the named cache at the cache manager;

read locking the named cache at the cache manager;

sending an indication that the named cache is read locked from the cache manager to the local node;

receiving the indication that the named cache is read locked at the local node;

requesting the cache item from a master locking database;

receiving the request for the cache item;

sending the cache item to the local node;

receiving the cache item at the local node;

sending a release of the read lock of the named cache from the local node to the cache manager;

receiving the release of the read lock of the named cache at the cache manager;

and

releasing the read lock of the named cache.

17. (original) Computer executable software code transmitted as an information signal, the code for cache management, the code comprising:

code to request a database lock of a named cache;

code to lock the named cache and provide an indication that the named cache is locked;

code to acquire a local lock of the named cache at a local node;

code to generate a timestamp corresponding to the local lock;
code to invalidate a cache item of the named cache in the local node;
code to release the local lock of the named cache;
code to send a message to a remote node identifying the cache item of the named
cache;
code to receive acknowledgment from the remote node;
code to send an update of the cache item of the named cache;
code to update the named cache; and
code to release the database lock of the named cache.

18. (original) A computer readable medium having computer executable program
code stored thereon, the code for cache management, the code comprising:

code to request a database lock of a named cache;
code to lock the named cache and provide an indication that the named cache is
locked;
code to acquire a local lock of the named cache at a local node;
code to generate a timestamp corresponding to the local lock;
code to invalidate a cache item of the named cache in the local node;
code to release the local lock of the named cache;
code to send a message to a remote node identifying the cache item of the named
cache;
code to receive acknowledgment from the remote node;
code to send an update of the cache item of the named cache;

code to update the named cache; and

code to release the database lock of the named cache.

19. (original) A programmed computer for cache management, comprising:

a memory having at least one region for storing computer executable program

code; and

a processor for executing the program code stored in memory, wherein the

program code comprises:

code to request a database lock of a named cache;

code to lock the named cache and provide an indication that the named cache is
locked;

code to acquire a local lock of the named cache at a local node;

code to generate a timestamp corresponding to the local lock;

code to invalidate a cache item of the named cache in the local node;

code to release the local lock of the named cache;

code to send a message to a remote node identifying the cache item of the named
cache;

code to receive acknowledgment from the remote node;

code to send an update of the cache item of the named cache;

code to update the named cache; and

code to release the database lock of the named cache.

20. (new) Computer executable software code transmitted as an information

signal, the code for cache management in a distributed computer system, the code comprising:

- code to identify a cache miss of a cache item;
- code to request a read lock of a named cache, the named cache including the cache item;
- code to read lock the named cache;
- code to request the cache item from a master locking database;
- code to receive the cache item; and
- code to release the read lock of the named cache.

21. (new) A computer readable medium having computer executable program code stored thereon, the code for cache management in a distributed computer system, the code comprising:

- code to identify a cache miss of a cache item;
- code to request a read lock of a named cache, the named cache including the cache item;
- code to read lock the named cache;
- code to request the cache item from a master locking database;
- code to receive the cache item; and
- code to release the read lock of the named cache.

22. (new) A programmed computer for cache management in a distributed computer system, comprising:

- a memory having at least one region for storing computer executable program

code; and

a processor for executing the program code stored in memory, wherein the
program code comprises:

code to identify a cache miss of a cache item;

code to request a read lock of a named cache, the named cache including the cache
item;

code to read lock the named cache;

code to request the cache item from a master locking database;

code to receive the cache item; and

code to release the read lock of the named cache.

23. (new) Computer executable software code transmitted as an information
signal, the code for cache management in a distributed computer system, the code comprising:

code to identify a cache miss of a cache item;

code to request a read lock of a global database, the global database including the
cache item;

code to read lock the global database;

code to request the cache item from a master locking database;

code to receive the cache item; and

code to release the read lock of the global database.

24. (new) A computer readable medium having computer executable program
code stored thereon, the code for cache management in a distributed computer system, the code

comprising:

- code to identify a cache miss of a cache item;
- code to request a read lock of a global database, the global database including the cache item;
- code to read lock the global database;
- code to request the cache item from a master locking database;
- code to receive the cache item; and
- code to release the read lock of the global database.

25. (new) A programmed computer for cache management in a distributed computer system, comprising:

- a memory having at least one region for storing computer executable program code; and
- a processor for executing the program code stored in memory, wherein the program code comprises:
 - code to identify a cache miss of a cache item;
 - code to request a read lock of a global database, the global database including the cache item;
 - code to read lock the global database;
 - code to request the cache item from a master locking database;
 - code to receive the cache item; and
 - code to release the read lock of the global database.

26. (new) Computer executable software code transmitted as an information signal, the code for cache management in a distributed computer system, the code comprising:

- code to determine that a predetermined event has occurred;
- code to request a read lock of a named cache;
- code to request a timestamp;
- code to receive an indication of a read lock of the named cache;
- code to receive a timestamp;
- code to compare the received timestamp with a previous timestamp;
- code to responsive to the comparison, perform a predetermined action; and
- code to release the read lock of the named cache.

27. (new) A computer readable medium having computer executable program code stored thereon, the code for cache management in a distributed computer system, the code comprising:

- code to determine that a predetermined event has occurred;
- code to request a read lock of a named cache;
- code to request a timestamp;
- code to receive an indication of a read lock of the named cache;
- code to receive a timestamp;
- code to compare the received timestamp with a previous timestamp;
- code to responsive to the comparison, perform a predetermined action; and
- code to release the read lock of the named cache.

28. (new) A programmed computer for cache management in a distributed computer system, comprising:

a memory having at least one region for storing computer executable program code; and

a processor for executing the program code stored in memory, wherein the program code comprises:

- code to determine that a predetermined event has occurred;
- code to request a read lock of a named cache;
- code to request a timestamp;
- code to receive an indication of a read lock of the named cache;
- code to receive a timestamp;
- code to compare the received timestamp with a previous timestamp;
- code to responsive to the comparison, perform a predetermined action; and
- code to release the read lock of the named cache.

29. (new) Computer executable software code transmitted as an information signal, the code for cache management, the code comprising:

code to determine at a local node that an update is required for a cache item in a named cache;

code to send a request for a database lock of the named cache from a local node to a cache manager;

code to receive the request for a database lock at the cache manager;

code to lock the database of the named cache;

code to send an indication that the database of the named cache is locked from the cache manager to the local node;

code to receive the indication that the database of the named cache is locked at the local node;

code to acquire at the local node a local write lock of the named cache;

code to send a timestamp from the local node to the cache manager;

code to receive the timestamp at the cache manager; updating a lock table with the timestamp;

code to invalidate at the local node the cache item of the named cache;

code to release the local write lock of the named cache;

code to send a message to invalidate the cache item of the named cache from the local node to a second node;

code to receive the message to invalidate at the second node acquiring at the second node a remote write lock of the named cache;

code to invalidate at the second node the cache item of the named cache;

code to send an acknowledgment of the message to invalidate from the second node to the local node;

code to release the remote write lock of the named cache at the second node;

code to receive the acknowledgment of the message to invalidate at the local node;

code to determine that no further acknowledgments of the message to invalidate are expected at the local node;

code to send an update of the cache item of the named cache from the local node;

code to receive the update of the cache item and updating the cache item; and
code to release the database lock of the named cache.

30. (new) A computer readable medium having computer executable program
code stored thereon, the code for cache management, the code comprising:

code to determine at a local node that an update is required for a cache item in a
named cache;

code to send a request for a database lock of the named cache from a local node to
a cache manager;

code to receive the request for a database lock at the cache manager;

code to lock the database of the named cache;

code to send an indication that the database of the named cache is locked from the
cache manager to the local node;

code to receive the indication that the database of the named cache is locked at the
local node;

code to acquire at the local node a local write lock of the named cache;

code to send a timestamp from the local node to the cache manager;

code to receive the timestamp at the cache manager; updating a lock table with the
timestamp;

code to invalidate at the local node the cache item of the named cache;

code to release the local write lock of the named cache;

code to send a message to invalidate the cache item of the named cache from the
local node to a second node;

code to receive the message to invalidate at the second node acquiring at the second node a remote write lock of the named cache;

code to invalidate at the second node the cache item of the named cache;

code to send an acknowledgment of the message to invalidate from the second node to the local node;

code to release the remote write lock of the named cache at the second node;

code to receive the acknowledgment of the message to invalidate at the local node;

code to determine that no further acknowledgments of the message to invalidate are expected at the local node;

code to send an update of the cache item of the named cache from the local node;

code to receive the update of the cache item and updating the cache item; and

code to release the database lock of the named cache.

31. (new) A programmed computer for cache management, comprising:

a memory having at least one region for storing computer executable program code; and

a processor for executing the program code stored in memory, wherein the program code comprises:

code to determine at a local node that an update is required for a cache item in a named cache;

code to send a request for a database lock of the named cache from a local node to a cache manager;

code to receive the request for a database lock at the cache manager;

code to lock the database of the named cache;

code to send an indication that the database of the named cache is locked from the cache manager to the local node;

code to receive the indication that the database of the named cache is locked at the local node;

code to acquire at the local node a local write lock of the named cache;

code to send a timestamp from the local node to the cache manager;

code to receive the timestamp at the cache manager; updating a lock table with the timestamp;

code to invalidate at the local node the cache item of the named cache;

code to release the local write lock of the named cache;

code to send a message to invalidate the cache item of the named cache from the local node to a second node;

code to receive the message to invalidate at the second node acquiring at the second node a remote write lock of the named cache;

code to invalidate at the second node the cache item of the named cache;

code to send an acknowledgment of the message to invalidate from the second node to the local node;

code to release the remote write lock of the named cache at the second node;

code to receive the acknowledgment of the message to invalidate at the local node;

code to determine that no further acknowledgments of the message to invalidate

are expected at the local node;

code to send an update of the cache item of the named cache from the local node;

code to receive the update of the cache item and updating the cache item; and

code to release the database lock of the named cache.

32. (new) Computer executable software code transmitted as an information signal, the code for cache management in a distributed computer system, the code comprising:

code to identify a cache miss of a cache item at a local node;

code to send a request for a read lock of a named cache from the local node to a cache manager;

code to receive the request for a read lock of the named cache at the cache manager;

code to read lock the named cache at the cache manager;

code to send an indication that the named cache is read locked from the cache manager to the local node;

code to receive the indication that the named cache is read locked at the local node;

code to request the cache item from a master locking database;

code to receive the request for the cache item;

code to send the cache item to the local node;

code to receive the cache item at the local node;

code to send a release of the read lock of the named cache from the local node to the cache manager;

code to receive the release of the read lock of the named cache at the cache manager; and

code to release the read lock of the named cache.

33. (new) A computer readable medium having computer executable program code stored thereon, the code for cache management in a distributed computer system, the code comprising:

code to identify a cache miss of a cache item at a local node;

code to send a request for a read lock of a named cache from the local node to a cache manager;

code to receive the request for a read lock of the named cache at the cache manager;

code to read lock the named cache at the cache manager;

code to send an indication that the named cache is read locked from the cache manager to the local node;

code to receive the indication that the named cache is read locked at the local node;

code to request the cache item from a master locking database;

code to receive the request for the cache item;

code to send the cache item to the local node;

code to receive the cache item at the local node;

code to send a release of the read lock of the named cache from the local node to the cache manager;

code to receive the release of the read lock of the named cache at the cache manager; and

code to release the read lock of the named cache.

34. (new) A programmed computer for cache management in a distributed computer system, comprising:

a memory having at least one region for storing computer executable program code; and

a processor for executing the program code stored in memory, wherein the program code comprises:

code to identify a cache miss of a cache item at a local node;

code to send a request for a read lock of a named cache from the local node to a cache manager;

code to receive the request for a read lock of the named cache at the cache manager;

code to read lock the named cache at the cache manager;

code to send an indication that the named cache is read locked from the cache manager to the local node;

code to receive the indication that the named cache is read locked at the local node;

code to request the cache item from a master locking database;

code to receive the request for the cache item;

code to send the cache item to the local node;

code to receive the cache item at the local node;

code to send a release of the read lock of the named cache from the local node to
the cache manager;

code to receive the release of the read lock of the named cache at the cache
manager; and

code to release the read lock of the named cache.
